JSC 38607

Correlation of the Rapid Development Methodology to the Software Engineering Institute's Capability Maturity Model

ENGINEERING DIRECTORATE

AEROSCIENCE AND FLIGHT MECHANICS DIVISION

15 July 1997

National Aeronautics and
Space Administration

Lyndon B. Johnson Space Center
Houston, TX

JSC 38607

Correlation of the Rapid Development Methodology to the Software Engineering Institute's Capability Maturity Model

Prepared By:

Denise M. DiFilippo
G. B. Tech, Incorporated

Approved By:

David A. Petri
GN&C Rapid Development Lab Manager
Aeroscience and Flight Mechanics Division
NASA/Johnson Space Center

James P. Ledet
Code Q RTOP Project Manager
Aeroscience and Flight Mechanics Division
NASA/Johnson Space Center

Concurred By:

Aldo J. Bordano, Chief
Aeroscience and Flight Mechanics Division
NASA/Johnson Space Center

Karen D. Frank, Chief
GN&C Development & Test Branch
Aeroscience and Flight Mechanics Division
NASA/Johnson Space Center

Table of Contents

Section *Page*

List of Tables

Acronyms and Abbreviations

AFMD	Aeroscience and Flight Mechanics Division
AFMD	Aeroscience and Flight Mechanics Division
COTS	commercial, off-the-shelf
CM	Configuration Management
CMM	Capability Maturity Model
DOF	degrees-of-freedom
GN&C	Guidance, Navigation & Control
HIL	hardware-in-the-loop
IV&V	Independent Verification and Validation
ISI	Integrated Systems Inc.
ISSA	International Space Station Alpha
JPL	Jet Propulsion Laboratory
JSC	Johnson Spaceflight Center
KPA	Key Process Area
MDA-HD	McDonnell Douglas Aerospace – Houston Division
MOD	Mission Operations Directorate
NASA	National Aeronautics and Space Administration
OIL	Operator-in-the-loop
QA	Quality Assurance
RD	Rapid Development
RDL	Rapid Development Laboratory
RTOP	Research and Technology Objectives and Plan
SCM	Software Configuration Management
SEI	Software Engineering Institute
SOW	Statement of Work

1.0 Introduction

The Aeroscience and Flight Mechanics Division (AFMD) at the National Aeronautics and Space Administration-Johnson Space Center (NASA-JSC) Engineering Directorate is exploring ways of producing Guidance, Navigation & Control (GN&C) systems more efficiently and effectively. A significant portion of this effort is software development, integration, testing and verification.

To achieve these goals, the AFMD established the GN&C Rapid Development Laboratory (RDL), a hardware/software facility designed to take a GN&C design project from initial inception through hardware-in-the-loop (HIL) testing and perform final GN&C system verification. The operations approach for the RDL concentrated on the use of commercial, off-the-shelf (COTS) software products to develop the GN&C algorithms in the form of graphical data flow diagrams, to automatically generate source code from these diagrams and to run in a real-time, HIL environment under a Rapid Development paradigm.

The success of these efforts motivated further study and documentation of Rapid Development methodologies. The initial goal was to formalize the successful methods used to date in the GN&C RDL. Subsequently the team expanded on these methods, based on knowledge gained from extensive search and study of the current literature. The resulting methodology is documented in JSC-38605, *Guidelines for the Rapid Development of Software Systems*. Using these guidelines as a baseline, study of Rapid Development as a formal methodology continues.

As part of this ongoing study, the team has been observing and participating in a test project for several months. The test project is the GN&C Orbiter Upgrades Deorbit Phase Flight Software Demonstration Project. This project was planned to investigate the ability of the Aeroscience and Flight Mechanics Division, and the Rapid Development Laboratory, to quickly and effectively design, implement, test and deliver high quality GN&C flight software. In addition, it expands on the knowledge base of the RDL by implementing a flight phase not previously developed in the RDL.

The state of the art in software development today often emphasizes quality and capability standards. One such standard that is widely accepted is the Software Engineering Institute's (SEI) Capability Maturity Model (CMM) for Software. (SEI has published several reports describing the CMM and its use; see, for example, CMU/SEI-93-TR-24, *Capability Maturity Model*, which was a primary reference for this report.)

This report presents a summary of the software CMM and examines it in light of the Rapid Development (RD) methodology presented in JSC-38605. The correlation between CMM and RD is discussed both on a theoretical level and as observed in the current test project.

2.0 Key Process Areas by Maturity Level

There are five major capability levels defined by the CMM, from least mature (level 1, Performed Informally) to most mature (level 5, Continuously Improving). Associated with maturity levels two through five are Key Process Areas (KPAs). These KPAs define the capabilities that characterize the associated maturity level. The Maturity Levels and their associated KPAs are shown in the table below. Naming terminology for the levels is not consistent throughout the literature. Two common names for each of the levels are given in the table. The levels are also referred to numerically, with level 1 being the least mature (corresponds to Initial), and level 5 the most mature (corresponds to Optimizing). Some of the literature also refers to level 0, sometimes named Not Performed or Ad Hoc, referring to organizations where development proceeds without any structured processes.

Table 1. Key Process Areas by Maturity Level

Maturity Level	Key Process Areas
Initial (Performed Informally)	(none)
Repeatable (Planned and Tracked)	Requirements Management
	Software Project Planning
	Software Project Tracking and Oversight
	Software Subcontract Management
	Software Quality Assurance
	Software Configuration Management
Defined (Well Defined)	Organization Process Focus
	Organization Process Definition
	Training Program
	Integrated Software Management
	Software Product Engineering
	Intergroup Coordination
	Peer Reviews
Managed (Quantitatively Controlled)	Quantitative Process Management
	Software Quality Management
Optimizing (Continuously Improving)	Defect Prevention
	Technology Change Management
	Process Change Management

3.0 Common Features of Key Process Areas

While each KPA has unique goals, they share several common features. These, along with some key characteristics of the common features, are shown in the table below.

Table 2.   Common Features

Common Features	Characterized by:
Commitment to Perform	Policy Statements
	Leadership
Ability to Perform	Resources and Funding
	Training
	Orientation
	Prerequisite Items
Activities Performed	
Measurement and Analysis	
Verifying Implementation	

4.0 About the Correlations

The following sections include tables which summarize the goals and key elements of the common features for each of the Key Practice Areas (KPAs), grouped by capability level. For each of levels 2 (repeatable) and 3 (defined), a discussion follows the tables and examines the key areas of concern when using the Rapid Development model for software development.

Overall, the Rapid Development guidelines are currently less mature, less complete, and less well tested than the CMM. Given the current maturity level of the RD model, it would be of questionable value, at this time, to evaluate the guidelines with respect to CMM levels 4 and 5, and no such evaluation is attempted here.

The Rapid Development life-cycle emphasizes software development techniques; the CMM emphasizes software project control, management and quality processes. Therefore, a direct correlation is not appropriate. What we have done is to look for and identify areas of inconsistency, concern, special interest, or in need of additional study between the CMM and RD. Many of the KPA goals and features invoke no special treatment when using Rapid Development, and these are included in the tables without comment.

In some cases, experiences with the current test project have revealed particular insight regarding an aspect of the CMM as relates to Rapid Development. These are included in the discussions.

For levels 2 and 3, the tables include columns for “theory” and “test”. An “x” in the “Theory” column implies that no specific Rapid Development issues have been identified relative to the goal or feature. An “x” in the “Test” column implies that the test project has complied with the goal or feature. When there is a numerical entry in the column, special characteristics have been identified relative to Rapid Development, or the test project has not complied with that item, and these will be included in the discussion sections.

5.0 Rapid Development Correlation with CMM Level 2 KPAs

In order to have reached capability maturity level 2 (Repeatable) an organization must have demonstrated capability in six Key Process Areas (Requirements Management, Software Project Planning, Software Project Tracking and Oversight, Software Subcontract Management, Software Quality Assurance, and Software Configuration Management). Tables 3 through 8 show the goals and key elements of the Common Features for each of these level 2 KPAs.

When comparing these tables to the Rapid Development guidelines, several general topics were identified for discussion. The relevant areas of the table are identified by numbers in the columns labelled “Theory” and “Test”, and a discussion of each topic follows the tables, in correspondingly numbered subsections (For example, all activities numbered “1” in the tables will be discussed as a group in section 5.1.).

Table 3. Requirements Management

		Theory	Test
Goals	System requirements allocated to software are controlled to establish a baseline for software engineering and management use	x	x
	Software plans, products, and activities are kept consistent with the system requirements allocated to software	x	x
Commitment to Perform	The project follows a written organizational policy for managing the system requirements allocated to software	x	6
Ability to Perform	For each project, responsibility is established for analyzing the system requirements and allocating them to hardware, software, and other system components	x	x
	The allocated requirements are documented	x	x
	Adequate resources and funding are provided for managing the allocated requirements	x	x
	Members of the software engineering group and other software-related groups are trained to perform their requirements management activities	x	x
Activities performed	The software engineering group reviews the allocated requirements before they are incorporated into the software project	x	x
	The software engineering group uses the allocated requirements as the basis for software plans, work products, and activities	x	x
	Changes to the allocated requirements are reviewed and incorporated into the software project	x	x
Measurement and Analysis	Measurements are made and used to determine the status of the activities for managing the allocated requirements	1	x
Verifying Implementation	The activities for managing the allocated requirements are reviewed with senior management on a periodic basis	x	x
	The activities for managing the allocated requirements are reviewed with the project manager on both a periodic and event-driven basis	x	x
	The software quality assurance group reviews and/or audits the activities and work products for managing the allocated requirements and reports the results	x	7

Table 4. Software Project Planning

		Theory	Test
Goals	Software estimates are documented for use in planning and tracking the software project	x	x
	Software project activities and commitments are planned and documented	x	x
	Affected groups and individuals agree to their commitments related to the software project	x	x
Commitment to Perform	A project software manager is designated to be responsible for negotiating commitments and developing the project’s software development plan	x	x
	The project follows a written organizational policy for planning a software project	x	6
Ability to Perform	A documented and approved statement of work exists for the software project	x	8
	Responsibilities for developing the software development plan are assigned	x	x
	Adequate resources and funding are provided for planning the software project	x	x
	The software managers, software engineers, and other individuals involved in the software project planning are trained in the software estimating and planning procedures applicable to their areas of responsibility	x	x
Activities performed	The software engineering group participates on the project proposal team	2	x
	Software project planning is initiated in the early stages of, and in parallel with, the overall project planning	2	x
	The software engineering group participates with other affected groups in the overall project planning throughout the project’s life	2	x
	Software project commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure	x	6
	A software life cycle with predefined stages of manageable size is identified or defined	3	x
	The project’s software development plan is developed according to a documented procedure	3	6
	The plan for the software project is documented	x	x

Table 4. Software Project Planning

		Theory	Test
	Software work products that are needed to establish and maintain control of the software project are identified	x	x
	Estimates for the size of the software work products (or changes to the size of software work products) are derived according to a documented procedure	4	x
	Estimates for the software project's effort and costs are derived according to a documented procedure	4	x
	Estimates for the project's critical computer resources are derived according to a documented procedure	4	x
	The project's software schedule is derived according to a documented procedure	4	x
	The software risks associated with the cost, resource, schedule, and technical aspects of the project are identified, assessed, and documented	4	x
	Plans for the project's software engineering facilities and support tools are prepared	x	x
	Software planning data are recorded	x	x
Measurement and Analysis	Measurements are made and used to determine the status of the software planning activities	1	x
Verifying Implementation	The activities for software project planning are reviewed with senior management on a periodic basis	x	x
	The activities for software project planning are reviewed with the project manager on both a periodic and event-driven basis	x	x
	The software quality assurance group reviews and/or audits the activities and work products for software project planning and reports the results	x	7

Table 5. Software Tracking and Oversight

		Theory	Test
Goals	Actual results and performances are tracked against the software plans	x	x
	Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans	x	x
	Changes to software commitments are agreed to by the affected groups and individuals	x	x
Commitment to Perform	A project software manager is designated to be responsible for the project's software activities and results	x	x
	The project follows a written organizational policy for managing the software project	x	6
Ability to Perform	A software development plan for the software project is documented and approved	x	x
	The project software manager explicitly assigns responsibility for software work products and activities	x	x
	Adequate resources and funding are provided for tracking the software project	x	x
	The software managers are trained in managing the technical and personnel aspects of the software project	x	x
	First-line software managers receive orientation in the technical aspects of the software project	x	x
Activities performed	A documented software development plan is used for tracking the software activities and communicating status	x	x
	The project's software development plan is revised according to a documented procedure	x	x
	Software project commitments and changes to commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure	x	x
	Approved changes to commitments that affect the software project are communicated to the members of the software engineering group and other software-related groups	x	x

Table 5. Software Tracking and Oversight

		Theory	Test
	The size of the software work products (or size of changes to the software work products) are tracked, and corrective actions are taken as necessary	4	x
	The project's software effort and costs are tracked, and corrective actions are taken as necessary	4	x
	The project's critical computer resources are tracked, and corrective actions are taken as necessary	x	x
	The project's software schedule is tracked, and corrective actions are taken as necessary	x	x
	Software engineering technical activities are tracked, and corrective actions are taken as necessary	x	x
	The software risks associated with cost, resource, schedule, and technical aspects of the project are tracked	4	8
	Actual measurement data and replanning data for the software project are recorded	x	x
	The software engineering group conducts periodic internal reviews to track technical progress, plans, performance, and issues against the software development plan	x	x
	Formal reviews to address the accomplishments and results of the software project are conducted at selected project milestones according to a documented procedure	x	x
Measurement and Analysis	Measurements are made and used to determine the status of the software tracking and oversight activities	1	x
Verifying Implementation	The activities for software project tracking and oversight are reviewed with senior management on a periodic basis	x	x
	The activities for software project tracking and oversight are reviewed with the project manager on both a periodic and event-driven basis	x	x
	The software quality assurance group reviews and/or audits the activities and work products for software project tracking and oversight and reports the results	x	7

Table 6. Software Subcontract Management

		Theory	Test
Goals	The prime contractor selects qualified software subcontractors	x	x
	The prime contractor and the software subcontractor agree to their commitments to each other	5	x
	The prime contractor and the software subcontractor maintain ongoing communications	x	9
	The prime contractor tracks the software subcontractor's actual results and performance against its commitment	5	9
Commitment to Perform	The project follows a written organizational policy for managing the software subcontract	5	9
	A subcontract manager is designated to be responsible for establishing and managing the software subcontract	x	9
Ability to Perform	Adequate resources and funding are provided for selecting the software subcontractor and managing the subcontract	x	9
	Software managers and other individuals who are involved in establishing and managing the software subcontract are trained to perform these activities	x	9
	Software managers and other individuals who are involved in managing the software subcontract receive orientation in the technical aspects of the subcontract	x	9
Activities performed	The work to be subcontracted is defined and planned according to a documented procedure	5	9
	The software subcontractor is selected, based on an evaluation of the subcontract bidders' ability to perform the work, according to a documented procedure	x	9
	The contractual agreement between the prime contractor and the software subcontractor is used as the basis for managing the subcontract	x	9
	A documented subcontractor's software development plan is reviewed and approved by the prime contractor	x	9
	A documented and approved subcontractor's software development plan is used for tracking the software activities and communicating status	x	9

Table 6. Software Subcontract Management

		Theory	Test
	Changes to the software subcontractor's statement of work, subcontract terms and conditions, and other commitments are resolved according to a documented procedure	5	9
	The prime contractor's management conducts periodic status/coordination reviews with the software subcontractor's management	x	9
	Periodic technical reviews and interchanges are held with the software subcontractor	x	9
	Formal reviews to address the subcontractor's software engineering accomplishments and results are conducted at selected milestones according to a documented procedure	x	9
	The prime contractor's software quality assurance group monitors the subcontractor's software quality assurance activities according to a documented procedure	x	9
	The prime contractor's software configuration management group monitors the subcontractor's activities for software configuration management according to a documented procedure	x	9
	The prime contractor conducts acceptance testing as part of the delivery of the subcontractor's software products according to a documented procedure	x	9
	The software subcontractor's performance is evaluated on a periodic basis, and the evaluation is reviewed with the subcontractor	x	9
Measurement and Analysis	Measurements are made and used to determine the status of the activities for managing the software subcontract	1	9
Verifying Implementation	The activities for managing the software subcontract are reviewed with senior management on a periodic basis	x	9
	The activities for managing the software subcontract are reviewed with the project manager on both a periodic and event-driven basis	x	9
	The software quality assurance group reviews and/or audits the activities and work products for managing the software subcontract and reports the results	x	9

Table 7. Software Quality Assurance

		Theory	Test
Goals	Software quality assurance activities are planned	x	7
	Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively	x	7
	Affected groups and individuals are informed of software quality assurance activities and results	x	7
	Noncompliance issues that cannot be resolved within the software project are addressed by senior management	x	7
Commitment to Perform	The project follows a written organizational policy for implementing software quality assurance (SQA)	x	7
Ability to Perform	A group that is responsible for coordinating and implementing SQA for the project (i.e., the SQA group) exists	x	7
	Adequate resources and funding are provided for performing the SQA activities	x	7
	Members of the SQA group are trained to perform their SQA activities	x	7
	The members of the software project receive orientation on the role, responsibilities, authority, and value of the SQA group	x	7
Activities Performed	A SQA plan is prepared for the software project according to a documented procedure	x	7
	The SQA group's activities are performed in accordance with the SQA plan	x	7
	The SQA group participates in the preparation and review of the project's software development plan, standards, and procedures	x	7
	The SQA group reviews the software engineering activities to verify compliance	x	7
	The SQA group audits designated software work products to verify compliance	x	7
	The SQA group periodically reports the results of its activities to the software engineering group	x	7
	Deviations identified in the software activities and software work products are documented and handled according to a documented procedure	x	7

Table 7. Software Quality Assurance

		Theory	Test
	The SQA group conducts periodic reviews of its activities and findings with the customer's SQA personnel, as appropriate	x	7
Measurement and Analysis	Measurements are made and used to determine the cost and schedule status of the SQA activities	1	7
Verifying Implementation	The SQA activities are reviewed with senior management on a periodic basis	x	7
	The SQA activities are reviewed with the project manager on both a periodic and event-driven basis	x	7
	Experts independent of the SQA group periodically review the activities and software work products of the project's SQA group	x	7

Table 8. Software Configuration Management

		Theory	Test
Goals	Software configuration management activities are planned	x	x
	Selected software work products are identified, controlled, and available	x	x
	Changes to identified software work products are controlled	x	x
	Affected groups and individuals are informed of the status and content of software baselines	x	x
Commitment to Perform	The project follows a written organizational policy for implementing software configuration management (SCM)	x	10
Ability to Perform	A board having the authority for managing the project's software baselines (i.e., a software configuration control board-SCCB) exists or is established	x	x
	A group that is responsible for coordinating and implementing SCM for the project (i.e., the SCM group) exists	x	x
	Adequate resources and funding are provided for performing the SCM activities	x	x
	Member of the SCM group are trained in the objectives, procedures, and methods for performing their SCM activities	x	x
	Members of the software engineering group and other software-related groups are trained to perform their SCM activities	x	x
Activities Performed	A SCM plan is prepared for each software project according to a documented procedure	x	10
	A documented and approved SCM plan is used as the basis for performing the SCM activities	x	x
	A configuration management library system is established as a repository for the software baselines	x	x
	The software work products to be placed under configuration management are identified	x	10
	Change requests and problem reports for all configuration items/units are initiated, recorded, reviewed, approved, and tracked according to a documented procedure	x	x

Table 8. Software Configuration Management

		Theory	Test
	Changes to baselines are controlled according to a documented procedure	x	x
	Products from the software baseline library are created and their release is controlled according to a documented procedure	x	x
	The status of configuration items/units is recorded according to a documented procedure	x	10
	Standard reports documenting the SCM activities and the contents of the software baseline are developed and made available to affected groups and individuals	x	10
	Software baseline audits are conducted according to a documented procedure	x	10
Measurement and Analysis	Measurements are made and used to determine the status of the SCM activities	1	x
Verifying Implementation	The SCM activities are reviewed with senior management on a periodic basis	x	8
	The SCM activities are reviewed with the project manager on both a periodic and event-driven basis	x	x
	The SCM group periodically audits software baselines to verify that they conform to the documentation that defines them	x	8
	The software quality assurance group reviews and/or audits the activities and work products for SCM and reports the results	x	7

For reference, the bullets in each discussion subsection include the related activity descriptions (from the tables).

5.1 Measurement

- Measurements are made and used to determine the status of the activities for managing the allocated requirements
- Measurements are made and used to determine the status of the software planning activities
- Measurements are made and used to determine the status of the software tracking and oversight activities
- Measurements are made and used to determine the status of the activities for managing the software subcontract
- Measurements are made and used to determine the cost and schedule status of the SQA activities
- Measurements are made and used to determine the status of the SCM activities

What are the appropriate metrics to measure a project when using Rapid Development techniques? It is anticipated that these will differ somewhat from metrics for more traditional projects.

For example, when using graphical programming tools (for example, MATRIXx), lines of code measurements may not be particularly useful. Another example is requirements tracking measurements, which will likely require significantly different metrics in a development environment that includes requirements evolution as a planned part of the life-cycle.

As part of this research, a program is in place for identifying, collecting and evaluating metrics in a Rapid Development environment.

5.2 Enablers

- The software engineering group participates on the project proposal team
- Software project planning is initiated in the early stages of, and in parallel with, the overall project planning
- The software engineering group participates with other affected groups in the overall project planning throughout the project's life

The Rapid Development life-cycle will strongly encourage compliance with the CMM in these areas. This is as a result of its emphasis on the core team and user involvement, as well as the concurrent engineering aspects of the method.

5.3 Focus Change

- A software life cycle with predefined stages of manageable size is identified or defined
- The project's software development plan is developed according to a documented procedure

These items can certainly be satisfied when using Rapid Development; they are called out here to accent awareness about the ways that they might be different from traditional development approaches.

When working in a Rapid Development environment, life cycle stages are typically not the same as for traditional waterfall based development methodologies. In fact, these stages may be tailored for each project.

Software development plans for Rapid Development should be expected to be somewhat more volatile than traditional plans. This is because the various cycles are planned to include discovery of information needed to complete plans for successive cycles.

5.4 Project Estimation

- Estimates for the size of the software work products (or changes to the size of software work products) are derived according to a documented procedure
- Estimates for the software project's effort and costs are derived according to a documented procedure
- Estimates for the project's critical computer resources are derived according to a documented procedure
- The project's software schedule is derived according to a documented procedure
- The software risks associated with the cost, resource, schedule, and technical aspects of the project are identified, assessed, and documented

Current project estimation technology does not readily and effectively produce accurate cost and scheduling estimates for projects developed using the Rapid Development life cycle. Additional research is needed, to determine appropriate costing parameters as well as cost and schedule algorithms. A proposal has been submitted to begin this work.

5.5 Areas of Uncertainty

- The prime contractor and the software subcontractor agree to their commitments to each other
- The prime contractor tracks the software subcontractor's actual results and performance against its commitment
- The project follows a written organizational policy for managing the software subcontract
- The work to be subcontracted is defined and planned according to a documented procedure
- Changes to the software subcontractor's statement of work, subcontract terms and conditions, and other commitments are resolved according to a documented procedure

Our experiences to date, with subcontract management when using the Rapid Development methodology, are limited to relatively small groups with frequent opportunities to communicate and work with each other. From our observations, it can reasonably be assumed that the Rapid Development methodology will add some complications to subcontract management.

For example, requirements discovery continues well into the development cycles under evolutionary Rapid Development. This will likely have repercussions on the statement of work

(SOW), schedules and costs. Subcontractor commitments and prime contractor tracking and evaluation of performance will require careful evaluation.

The teaming arrangements used to accomplish the work is critical in a Rapid Development environment. The flexibility to respond quickly newly acquired data, information, and results, must be built into subcontract management procedures. SOWs must be flexible, and to contain the associated risk, the subcontract manager may be required to be a more active team member than is typical under traditional methodologies.

5.6 Written Policies

- The project follows a written organizational policy for managing the system requirements allocated to software
- The project follows a written organizational policy for planning a software project
- Software project commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure
- The project's software development plan is developed according to a documented procedure
- The project follows a written organizational policy for managing the software project
- The project's software development plan is revised according to a documented procedure
- Software project commitments and changes to commitments made to individuals and groups external to the organization are reviewed with senior management according to a documented procedure

These are called out for completeness. They do not represent problems unique to Rapid Development, and the actions noted were performed for the test project. The project team did not, however, have or create written policies and procedures for these activities.

5.7 Quality Assurance Group

- The software quality assurance group reviews and/or audits the activities and work products for managing the allocated requirements and reports the results
- The software quality assurance group reviews and/or audits the activities and work products for software project planning and reports the results
- The software quality assurance group reviews and/or audits the activities and work products for software project tracking and oversight and reports the results
- The software quality assurance group reviews and/or audits the activities and work products for SCM and reports the results
- plus all of Table 7 (Software Quality Assurance)

The scope of this demonstration project did not expand to include formal Quality Assurance (QA) functions during initial development. The project plans do include Independent Verification and Validation (IV&V), planned to occur after integration testing is complete, to assist in evaluation of the project.

There is no evidence that Rapid Development would impose any undue strain on traditional

QA functions, though a test at a later time would be appropriate. The project team has discussed the issue, and most of the questions which were raised concerned what level of QA would be appropriate in the early cycles of the evolutionary development model.

The project team is experimenting with new ways to inject quality control into the development process, besides the traditional, independent, QA group and functions. A technique that is showing great promise, and is consistent with the Rapid Development core team and concurrent engineering concepts, involves embedding QA functions and tools into the development process and the team. Use of analytical, automated Quality Assurance tools can accelerate QA evaluation and provide information to the developers quickly. Some advanced tools can be used directly by the developers, allowing them to evaluate and improve the quality of software as it is being developed.

5.8 Not Done

- A documented and approved statement of work exists for the software project
- The software risks associated with cost, resource, schedule, and technical aspects of the project are tracked
- The SCM activities are reviewed with senior management on a periodic basis
- The SCM group periodically audits software baselines to verify that they conform to the documentation that defines them

These activities were not performed during the test project.

The risk assessment activity is predicated on project estimation capability which, as previously mentioned, is still under investigation for Rapid Development.

The Configuration Management activities should be easy enough to implement once the CM system has been completely implemented (see section 5.10 below).

The statement of work evolved by consensus within the project team, but we neglected to take the final step of formally having it approved. Since the team moved very quickly to functional requirements, and these were documented and approved, there was very little risk incurred. In the future, if CMM compliance is desired, this activity would require additional attention, but no unique RD impacts are known.

5.9 Not Applicable

- All of Table 6 (Software Subcontract Management)

Both the core and expanded project teams, for the test project, included representatives of multiple contracting organizations. Yet, the project did not really address any significant subcontract management issues. This is because the team functioned as one working unit, under one project manager, and with common goals, schedules and deliverables, irrespective of any individual’s specific employer.

Thus the test project did not teach any significant lessons relative to Rapid Development compliance with the CMM subcontract management KPA.

5.10 In Work

- The project follows a written organizational policy for implementing software configuration management (SCM)
- A SCM plan is prepared for each software project according to a documented procedure
- The software work products to be placed under configuration management are identified
- The status of configuration items/units is recorded according to a documented procedure
- Standard reports documenting the SCM activities and the contents of the software baseline are developed and made available to affected groups and individuals
- Software baseline audits are conducted according to a documented procedure

The experiences of the test project team have shown that Rapid Development imposes some new requirements onto the traditional Configuration Management processes. The team is currently implementing and testing a possible CM approach. In future projects, CM will be imposed on the evolving software earlier in development.

Some of the questions which have surfaced center on the various cycles, both of delivered software and of documents. How many cycles should be maintained by CM? Which delivery is the first to be captured by the CM system? Can we devise a CM system which will adequately protect the project without sacrificing the momentum of evolutionary Rapid Development?

5.11 Summary for Level 2

There is nothing inherent in the Rapid Development Methodology that would prohibit an organization following it to reach level 2 CMM. There are likely to be some significant differences in some processes. And some areas require more research before they can be effectively implemented, especially metrics collection and evaluation, and project estimation techniques.

The practices which were used to perform the test project complied with most of the level 2 requirements. In addition to the research areas mentioned above, creation of certain written policy statements, and involvement of a quality assurance organization, are the primary areas of deficiency. These last two are general deficiencies not specifically related to Rapid Development.

6.0 Rapid Development Correlation with CMM Level 3 KPAs

In order to have reached capability maturity level 3 (Defined) an organization must have demonstrated capability in seven Key Process Areas (Organization Process Focus, Organization Process Definition, Training Program, Integrated Software Management, Software Product Engineering, Intergroup Coordination, and Peer Reviews) in addition to the six level 2 Key Process Areas (Requirements Management, Software Project Planning, Software Project Tracking and Oversight, Software Subcontract Management, Software Quality Assurance, and Software Configuration Management). Tables 9 through 15 show the goals and key elements of the Common Features for each of these level 2 KPAs.

When comparing these tables to the Rapid Development guidelines, several general topics were identified for discussion. The relevant areas of the table are identified by numbers in the columns labelled “Theory” and “Test”, and a discussion of each topic follows the tables, in correspondingly numbered subsections (For example, all activities numbered “1” in the tables will be discussed as a group in section 6.1.).

Table 9. Organization Process Focus

		Theory	Test
Goals	Software process development and improvement activities are coordinated across the organization	x	1
	The strengths and weaknesses of the software processes used are identified relative to a process standard	x	x
	Organization-level process development and improvement activities are planned	x	1
Commitment to Perform	The organization follows a written organizational policy for coordinating software process development and improvement activities across the organization	x	2
	Senior management sponsors the organization’s activities for software process development and improvement	x	x
	Senior management oversees the organization’s activities for software process development and improvement	x	x
Ability to Perform	A group that is responsible for the organization’s software process activities exists	x	x
	Adequate resources and funding are provided for the organization’s software process activities	x	x
	Members of the group responsible for the organization’s software process activities receive required training to perform these activities	x	x
	Members of the software engineering group and other software-related groups receive orientation on the organization’s software process activities and their roles in those activities	x	x
Activities Performed	The software process is assessed periodically, and action plans are developed to address the assessment findings	x	x
	The organization develops and maintains a plan for its software process development and improvement activities	x	2
	The organization’s and projects’ activities for developing and improving their software processes are coordinated at the organization level	x	x
	The use of the organizations software process database is coordinated at the organizational level	x	1

Table 9. Organization Process Focus

		Theory	Test
	New processes, methods, and tools in limited use in the organization are monitored, evaluated, and, where appropriate, transferred to other parts of the organization	x	3
	Training for the organization's and projects' software processes is coordinated across the organization	x	1
	The groups involved in implementing the software processes are informed of the organizations and projects' activities for software process development and improvement	x	x
Measurement and Analysis	Measurements are made and used to determine the status of the organization's process development and improvement activities	x	4
Verifying Implementation	The activities for software process development and improvement are reviewed with senior management on a periodic basis	x	x

Table 10. Organizing Process Definition

		Theory	Test
Goals	A standard software process for the organization is developed and maintained	x	1
	Information related to the use of the organization's standard software process by the software projects is collected, reviewed, and made available	x	1
Commitment to Perform	The organization follows a written policy for developing and maintaining a standard software process and related process assets	x	2
Ability to Perform	Adequate resources and funding are provided for developing and maintaining the organizations standard software process and related process assets	x	x
	The individuals who develop and maintain the organizations standard software process and related process assets receive required training to perform these activities	x	x
Activities Performed	The organization's standard software process is developed and maintained according to a documented procedure	x	2
	The organization's standard software process is documented according to established organization standards	x	1
	Descriptions of software life cycles that are approved for use by the projects are documented and maintained	x	x
	Guidelines and criteria for the projects' tailoring of the organization's standard software process are developed and maintained	x	1
	The organizations software process database is established and maintained	x	1
	A library of software process-related documentation is established and maintained	x	1
Measurement and Analysis	Measurements are made and used to determine the status of the organization's process definition activities	x	4
Verifying Implementation	The software quality assurance group reviews and/or audits the organization's activities and work products for developing and maintaining the organization's standard software process and related process assets and reports the results	x	5

Table 11. Training Program

		Theory	Test
Goals	Training activities are planned	x	x
	Training for developing the skills and knowledge needed to perform software management and technical roles is provided	x	x
	Individuals in the software engineering group and software-related groups receive the training necessary to perform their roles	x	x
Commitment to Perform	The organization follows a written policy for meeting its training needs	x	2
Ability to Perform	A group responsible for fulfilling the training needs of the organization exists	x	6
	Adequate resources and funding are provided for implementing the training program	x	x
	Members of the training group have the necessary skills and knowledge to perform their training activities	x	x
	Software managers receive orientation on the training program	x	x
Activities Performed	Each software project develops and maintains a training plan that specifies its training needs	x	6
	The organization’s training plan is developed and revised according to a documented procedure	x	2
	The training for the organization is performed in accordance with the organization’s training plan	x	6
	Training courses prepared at the organization level are developed and maintained according to organization standards	x	6
	A waiver procedure for required training is established and used to determine whether individuals already possess the knowledge and skills required to perform in their designated roles	x	6
	Records of training are maintained	x	6
Measurement and Analysis	Measurements are made and used to determine the status of the training program activities	4	x

Table 11. Training Program

		Theory	Test
	Measurements are made and used to determine the quality of the training program	4	x
Verifying Implementation	The training program activities are reviewed with senior management on a periodic basis	x	6
	The training program is independently evaluated on a periodic basis for consistency with, and relevance to, the organization’s needs	x	6
	The training program activities and work products are reviewed and/or audited and the results are reported	x	6

Table 12. Integrated Software Management

		Theory	Test
Goals	The project's defined software process is a tailored version of the organization's standard software process	x	x
	The project is planned and managed according to the project's defined software process	x	x
Commitment to Perform	The project follows a written organizational policy requiring that the software project be planned and managed using the organization's standard software process and related process assets	x	2
Ability to Perform	Adequate resources and funding are provided for managing the software project using the project's defined software process	x	x
	The individuals responsible for developing the project's defined software process receive required training in how to tailor the organization's standard software process and use the related process assets	x	1
	The software managers receive required training in managing the technical, administrative, and personnel aspects of the software project based on the project's defined software process	x	x
Activities Performed	The project's defined software process is developed by tailoring the organization's standard software process according to a documented procedure	x	2
	Each project's defined software process is revised according to a documented procedure	x	2
	The project's software development plan, which describes the use of the project's defined software process, is developed and revised according to a documented procedure	x	2
	The software project is managed in accordance with the project's defined software process	x	x
	The organization's software process database is used for software planning and estimating	x	1
	The size of the software work products (or size of changes to the software work products) is managed according to a documented procedure	x	2
	The project's software effort and costs are managed according to a documented procedure	x	2

Table 12. Integrated Software Management

		Theory	Test
	The project's critical computer resources are managed according to a documented procedure	x	2
	The critical dependencies and critical paths of the project's software schedule are managed according to a documented procedure	x	2
	The project's software risks are identified, assessed, documented, and managed according to a documented procedure	x	2
	Reviews of the software project are periodically performed to determine the actions needed to bring the software project's performance and results in line with the current and projected needs of the business, customer, and end users, as appropriate	x	x
Measurement and Analysis	Measurements are made and used to determine the effectiveness of the integrated software management activities	4	x
Verifying Implementation	The activities for managing the software project are reviewed with senior management on a periodic basis	x	x
	The activities for managing the software project are reviewed with the project manager on both a periodic and event-driven basis	x	x
	The software quality assurance group reviews and/or audits the activities and work products for managing the software project and reports the results	x	5

Table 13. Software Product Engineering

		Theory	Test
Goals	The software engineering tasks are defined, integrated, and consistently performed to produce the software	x	x
	Software work products are kept consistent with each other	x	x
Commitment to Perform	The project follows a written organizational policy for performing the software engineering activities	x	2
Ability to Perform	Adequate resources and funding are provided for performing the software engineering tasks	x	x
	Members of the software engineering technical staff receive required training to perform their technical assignments	x	x
	Members of the software engineering technical staff receive orientation in related software engineering disciplines	x	x
	The project manager and all software managers receive orientation in the technical aspects of the software project	x	x
Activities Performed	Appropriate software engineering methods and tools are integrated into the project's defined software process	x	x
	The software requirements are developed, maintained, documented, and verified by systematically analyzing the allocated requirements according to the project's defined software process	x	7
	The software design is developed, maintained, documented, and verified, according to the project's defined software process, to accommodate the software requirements and to form the framework for coding	x	7
	The software code is developed, maintained, documented, and verified, according to the project's defined software process, to implement the software requirements and software design	x	7
	Software testing is performed according to the project's defined software process	x	8
	Integration testing of the software is planned and performed according to the project's defined software process	x	8

Table 13. Software Product Engineering

		Theory	Test
	System and acceptance testing of the software are planned and performed to demonstrate that the software satisfies its requirements	x	x
	The documentation that will be used to operate and maintain the software is developed and maintained according to the project's defined software process	x	x
	Data on defects identified in peer reviews and testing are collected and analyzed according to the project's defined software process	x	x
	Consistency is maintained across software work products, including the software plans, process descriptions, allocated requirements, software requirements, software design, code, test plans, and test procedures	x	x
Measurement and Analysis	Measurements are made and used to determine the functionality and quality of the software products	4	x
	Measurements are made and used to determine the status of the software product engineering activities	4	x
Verifying Implementation	The activities for software product engineering are reviewed with senior management on a periodic basis	x	x
	The activities for software product engineering are reviewed with the project manager on both a periodic and event-driven basis	x	x
	The software quality assurance group reviews and/or audits the activities and work products for software product engineering and reports the results	x	5

Table 14. Intergroup Coordination

		Theory	Test
Goals	The customer's requirements are agreed to by all affected groups	x	x
	The commitments between the engineering groups are agreed to by the affected groups	x	x
	The engineering groups identify, track, and resolve intergroup issues	x	x
Commitment to Perform	The project follows a written organizational policy for establishing interdisciplinary engineering teams	x	2
Ability to Perform	Adequate resources and funding are provided for coordinating the software engineering activities with other engineering groups	x	x
	The support tools used by the different engineering groups are compatible to enable effective communication and coordination	x	x
	All managers in the organization receive required training in teamwork	x	x
	All task leaders in each engineering group receive orientation in the processes, methods, and standards used by the other engineering groups	x	x
	The members of the engineering groups receive orientation in working as a team	x	x
Activities Performed	The software engineering group and the other engineering groups participate with the customer and end users, as appropriate, to establish the system requirements	x	x
	Representatives of the project's software engineering group work with representatives of the other engineering groups to monitor and coordinate technical activities and resolve technical issues	x	x
	A documented plan is used to communicate intergroup commitments and to coordinate and track the work performed	x	2
	Critical dependencies between engineering groups are identified, negotiated, and tracked according to a documented procedure	x	2

Table 14. Intergroup Coordination

		Theory	Test
	Work products produced as input to other engineering groups are reviewed by representatives of the receiving groups to ensure that the work products meet their needs	x	x
	Intergroup issues not resolvable by the individual representatives of the project engineering groups are handled according to a documented procedure	x	2
	Representatives of the project engineering groups conduct periodic technical reviews and interchanges	x	x
Measurement and Analysis	Measurements are made and used to determine the status of the intergroup coordination activities	4	x
Verifying Implementation	The activities for intergroup coordination are reviewed with senior management on a periodic basis	x	x
	The activities for intergroup coordination are reviewed with the project manager on both a periodic and event-driven basis	x	x
	The software quality assurance group reviews and/or audits the activities and work products for intergroup coordination and reports the results	x	x

Table 15. Peer Reviews

		Theory	Test
Goals	Peer review activities are planned	x	x
	Defects in the software work products are identified and removed	x	x
Commitment to Perform	The project follows a written organizational policy for performing peer reviews	x	2
Ability to Perform	Adequate resources and funding are provided for performing peer reviews on each software work product to be reviewed	x	x
	Peer review leaders receive required training in how to lead peer reviews	x	x
	Reviewers who participate in peer reviews receive required training in the objectives, principles, and methods of peer reviews	x	x
Activities Performed	Peer reviews are planned, and the plans are documented	x	x
	Peer reviews are performed according to a documented procedure	x	x
	Data on the conduct and results of the peer reviews are recorded	x	x
Measurement and Analysis	Measurements are made and used to determine the status of the peer review activities	x	x
Verifying Implementation	The software quality assurance group reviews and/or audits the activities and work products for peer reviews and reports the results	x	5

For reference, the bullets in each discussion subsection include the related activity descriptions (from the tables).

6.1 Maturity Issues

- Software process development and improvement activities are coordinated across the organization
- Organization-level process development and improvement activities are planned
- The use of the organizations software process database is coordinated at the organizational level
- Training for the organization’s and projects’ software processes is coordinated across the organization
- A standard software process for the organization is developed and maintained
- Information related to the use of the organization’s standard software process by the software projects is collected, reviewed, and made available
- The organization’s standard software process is documented according to established organization standards
- Guidelines and criteria for the projects’ tailoring of the organization’s standard software process are developed and maintained
- The organizations software process database is established and maintained
- A library of software process-related documentation is established and maintained
- The individuals responsible for developing the project’s defined software process receive required training in how to tailor the organization’s standard software process and use the related process assets
- The organization’s software process database is used for software planning and estimating

Many of these organizational, institutionalization, and procedure issues ore being addressed by the RDL in its ISO-9000 compliance efforts, although they are not currently part of the test project. In theory, the Rapid Development process does not preclude compliance with these goals and activities. In practice, the technique is not mature enough to realistically and formally institutionalize these efforts.

As we continue to apply, observe, enhance, and mature the methodology, such things as a software process database, formalized tailoring and process improvement guidelines, process training programs, and organizational consistency, may more reasonably be expected to be put into place. The current research, of which this report is a part, is the initial seed to create the on-going improvement process.

For now, these activities are still goals more than reality. They are informally addressed, on a per project basis, as needed.

6.2 Written Procedures

The level 3 compliance requires the creation and adoption of many written organizational policies. Several of these, as noted in this section, were not available for the test project.

6.2.1 Written Procedures: Compliance Accessible with Current Methods

- The organization follows a written policy for meeting its training needs
- The organization’s training plan is developed and revised according to a documented procedure
- The project follows a written organizational policy requiring that the software project be planned and managed using the organization’s standard software process and related process assets
- Each project’s defined software process is revised according to a documented procedure
- The project’s software development plan, which describes the use of the project’s defined software process, is developed and revised according to a documented procedure
- The size of the software work products (or size of changes to the software work products) is managed according to a documented procedure
- The project’s software effort and costs are managed according to a documented procedure
- The project’s critical computer resources are managed according to a documented procedure
- The critical dependencies and critical paths of the project’s software schedule are managed according to a documented procedure
- The project follows a written organizational policy for performing the software engineering activities
- The project follows a written organizational policy for establishing interdisciplinary engineering teams
- A documented plan is used to communicate intergroup commitments and to coordinate and track the work performed
- Critical dependencies between engineering groups are identified, negotiated, and tracked according to a documented procedure
- Intergroup issues not resolvable by the individual representatives of the project engineering groups are handled according to a documented procedure
- The project follows a written organizational policy for performing peer reviews

These required written procedures either exist informally (and undocumented), are not specific to Rapid Development, or could reasonably be developed for Rapid Development at its current maturity level.

6.2.2 Written Procedures: Compliance Requires Further Maturity of Methods

- The organization follows a written organizational policy for coordinating software process development and improvement activities across the organization
- The organization develops and maintains a plan for its software process development and improvement activities
- The organization follows a written policy for developing and maintaining a standard software process and related process assets

- The organization’s standard software process is developed and maintained according to a documented procedure
- The project’s defined software process is developed by tailoring the organization’s standard software process according to a documented procedure
- The project’s software risks are identified, assessed, documented, and managed according to a documented procedure

These procedures relate to an organization’s standard software process, and process development and improvement activities. An overview of the standard Rapid Development process has been presented in the guidelines document (JSC-38605). The current test and research activities are aimed at validating and improving the process. In the RDL, an ISO-9000 Work Instruction for software development is being drafted. In these ways, standard software development and improvement procedures are evolving in our organization. If CMM level 3 compliance is required, preliminary written procedures could be produced based on our current understanding.

6.3 Enablers

- New processes, methods, and tools in limited use in the organization are monitored, evaluated, and, where appropriate, transferred to other parts of the organization

The Rapid Development life-cycle will strongly encourage compliance with the CMM in this areas. The study, evolution, and formalization of this new methodology is a large scale effort to propagate the use of successful new technology throughout the organization, agency, and industry.

6.4 Measurement

- Measurements are made and used to determine the status of the organization’s process development and improvement activities
- Measurements are made and used to determine the status of the organization’s process definition activities
- Measurements are made and used to determine the status of the training program activities
- Measurements are made and used to determine the quality of the training program
- Measurements are made and used to determine the effectiveness of the integrated software management activities
- Measurements are made and used to determine the functionality and quality of the software products
- Measurements are made and used to determine the status of the software product engineering activities
- Measurements are made and used to determine the status of the intergroup coordination activities

What are the appropriate metrics to measure a project when using Rapid Development techniques? It is anticipated that these will differ somewhat from metrics for more traditional projects.

As part of this research, a program is in place for identifying, collecting and evaluating metrics in a Rapid Development environment.

6.5 Quality Assurance Group

- The software quality assurance group reviews and/or audits the organization’s activities and work products for developing and maintaining the organization’s standard software process and related process assets and reports the results
- The software quality assurance group reviews and/or audits the activities and work products for managing the software project and reports the results
- The software quality assurance group reviews and/or audits the activities and work products for software product engineering and reports the results
- The software quality assurance group reviews and/or audits the activities and work products for peer reviews and reports the results

The scope of this demonstration project did not expand to include formal Quality Assurance (QA) functions during initial development.

Formal inspections (peer reviews) were held within the core team, but not audited by a quality assurance group. For those modules that were developed using a graphical programming environment (e.g., MATRIXx), inspections of the graphical representation were held, rather than the more traditional code inspection. The project plans also include Independent Verification and Validation (IV&V), planned to occur after integration testing is complete, to assist in evaluation of the project.

There is no evidence that Rapid Development would impose any undue strain on traditional QA functions, though a test at a later time would be appropriate. The project team has discussed the issue, and most of the questions which were raised concerned what level of QA would be appropriate in the early cycles of the evolutionary development model.

6.6 Not Done, or Done Informally

- A group responsible for fulfilling the training needs of the organization exists
- Each software project develops and maintains a training plan that specifies its training needs
- The training for the organization is performed in accordance with the organization’s training plan
- Training courses prepared at the organization level are developed and maintained according to organization standards
- A waiver procedure for required training is established and used to determine whether individuals already possess the knowledge and skills required to perform in their designated roles
- Records of training are maintained
- The training program activities are reviewed with senior management on a periodic basis
- The training program is independently evaluated on a periodic basis for consistency with, and relevance to, the organization’s needs

- The training program activities and work products are reviewed and/or audited and the results are reported

These activities were not formally performed during the test project. Training needs were based on self-evaluation by team members, and arranged by the project manager as need arose. Much of the required training was performed by knowledgeable team members who shared their knowledge with other team members. Generally this was handled through relatively short introductory presentations followed up with personal assistance as questions arose.

Procedurally, it would not be too difficult to comply with these training features. The flexible approach used by the project team proved to be quite effective and responsive to team needs. One reason it was so effective is that, in Rapid Development, the core team is constructed to include as many of the known required skills as possible. Another was the close working relationship among members of the core team, so that training needs were quickly identified and matched with available skills. Any formal process implemented should try to capture the advantages of this approach.

6.7 Focus Change

- The software requirements are developed, maintained, documented, and verified by systematically analyzing the allocated requirements according to the project’s defined software process
- The software design is developed, maintained, documented, and verified, according to the project’s defined software process, to accommodate the software requirements and to form the framework for coding
- The software code is developed, maintained, documented, and verified, according to the project’s defined software process, to implement the software requirements and software design

These items can certainly be satisfied when using Rapid Development; they are called out here to accent awareness about the ways that they might be different from traditional development approaches.

When working in a Rapid Development environment, requirements are usually not completely finished before development begins. Rather, requirements evolve as the system matures, understanding is improved, and needs are discovered. Therefore, the processes related to requirements are likely to continue throughout much of development.

When using a graphical development environment to enhance rapid development, especially if an autocoder is available, software design takes on a new emphasis. In fact, the design and implementation proceed concurrently. If the design is developed in the graphical language, and then autocoded, validating that code meets design may well be reduced to confidence levels in the autocoder. The issue of certification of autocoder output is of some consideration now, but this should become less problematic as autocoders mature and improve. Eventually, they may invoke a level of confidence similar to that afforded compilers today.

6.8 Test Strategies

- Software testing is performed according to the project’s defined software process
- Integration testing of the software is planned and performed according to the project’s defined software process

For the test project, there were many questions raised as to the appropriate testing strategy when using Rapid Development. The team settled on an approach that emphasized nominal case testing for early cycles, expanded unit testing to include off nominal cases as modules matured, and delayed full scale integration testing (to include of nominal cases) until a baseline has been completed and all modules had passed formal inspections.

Integration testing of the baseline is currently underway. IV&V is also planned. The team is investigating analytical tools that evaluate test coverage and software quality. Metrics gathered from these activities will assist in evaluating the test strategy.

6.9 Summary for Level 3

There is nothing inherent in the Rapid Development Methodology that would prohibit an organization following it to reach level 3 CMM. There are some areas where additional experience with and maturity of the Rapid Development model is required before the benefits of level 3 capability could be fully realized.

The practices which were used to perform the test project complied with many of the level 3 requirements. Quite a few formal, written procedures are required for full compliance.

7.0 Goals and Common Feature details for Level 4 KPAs

In order to have reached capability maturity level 4 (Managed) an organization must have demonstrated capability in two Key Process Areas (Quantitative Process Management and Software Quality Management) in addition to the six level 2 KPAs (Requirements Management, Software Project Planning, Software Project Tracking and Oversight, Software Subcontract Management, Software Quality Assurance, and Software Configuration Management) and the seven level three KPAs (Organization Process Focus, Organization Process Definition, Training Program, Integrated Software Management, Software Product Engineering, Intergroup Coordination, and Peer Reviews). The tables below show the goals and key features of the Common Features for each of these level 4 KPAs.

Table 16. Quantitative Process Management

Goals	The quantitative process management activities are planned
	The process performance of the project’s defined software process is controlled quantitatively
	The process capability of the organization’s standard software process is known in quantitative terms
Commitment to Perform	The project follows a written organizational policy for measuring and quantitatively controlling the performance of the project’s defined software process
	The organization follows a written policy for analyzing the process capability of the organization’s standard software process
Ability to Perform	A group that is responsible for coordinating the quantitative process management activities for the organization exists
	Adequate resources and funding are provided for the quantitative process management activities
	Support exists for collecting, recording, and analyzing data for selected process and product measurements
	The individuals implementing or supporting quantitative process management receive required training to perform these activities
	The members of the software engineering group and other software-related groups receive orientation on the goals and value of quantitative process management
Activities Performed	The software project’s plan for quantitative process management is developed according to a documented procedure
	The software project’s quantitative process management activities are performed in accordance with the project’s quantitative process management plan
	The strategy for the data collection and the quantitative analyses to be performed are determined based on the project’s defined software process
	The measurement data used to control the project’s defined software process quantitatively are collected according to a documented procedure
	The project’s defined software process in analyzed and brought under quantitative control according to a documented procedure
	Reports documenting the results of the software project’s quantitative process management activities are prepared and distributed

Table 16. Quantitative Process Management

	The process capability baseline for the organization’s standard software process is established and maintained according to a documented procedure
Measurement and Analysis	Measurements are made and used to determine the status of the activities for quantitative process management
Verifying Implementation	The activities for quantitative process management are reviewed with senior management on a periodic basis
	The software project’s activities for quantitative process management are reviewed with the project manager on both a periodic and event-driven basis
	The software quality assurance group reviews and/or audits the activities and work products for quantitative process management and reports the results

Table 17. Software Quality Management

Goals	The project’s software quality management activities are planned
	Measurable goals for software product quality and their priorities are defined
	Actual progress toward achieving the quality goals for the software products is quantified and managed
Commitment to Perform	The project follows a written organizational policy for managing software quality
Ability to Perform	Adequate resources and funding are provided for managing the quality of the software products
	The individuals implementing and supporting software quality management receive required training to perform their activities
	The members of the software engineering group and other software-related groups receive required training in software quality management
Activities Performed	The project’s software quality plan is developed and maintained according to a documented procedure
	The project’s software quality plan is the basis for the project’s activities for software quality management
	The project’s quantitative quality goals for the software products are defined, monitored, and revised throughout the software life cycle
	The quality of the project’s software products is measured, analyzed, and compared to the products’ quantitative quality goals on an event-driven basis
	The software project’s quantitative quality goals for the products are allocated appropriately to the subcontractors delivering software products to the project
Measurement and Analysis	Measurements are made and used to determine the status of the software quality management activities
Verifying Implementation	The activities for software quality management are reviewed with senior management on a periodic basis
	The activities for software quality management are reviewed with the project manager on both a periodic and event-drive basis
	The software quality assurance group reviews and/or audits the activities and work products for software quality management and reports the results

8.0 Goals and Common Feature details for Level 5 KPAs

In order to have reached capability maturity level 5 (Optimizing) an organization must have demonstrated capability in three Key Process Areas (Defect Prevention, Technology Change Management, and Process Change Management) in addition to the six level 2 KPAs (Requirements Management, Software Project Planning, Software Project Tracking and Oversight, Software Subcontract Management, Software Quality Assurance, and Software Configuration Management), the seven level three KPAs (Organization Process Focus, Organization Process Definition, Training Program, Integrated Software Management, Software Product Engineering, Intergroup Coordination, and Peer Reviews), and the two level 4 KPAs (Quantitative Process Management and Software Quality Management). The tables below show the goals and key features of the Common Features for each of these level 5 KPAs.

Table 18. Defect Prevention

Goals	Defect prevention activities are planned
	Common causes of defects are sought out and identified
	Common causes of defects are prioritized and systematically eliminated
Commitment to Perform	The organization follows a written policy for defect prevention activities
	The project follows a written organizational policy for defect prevention activities
Ability to Perform	An organization-level team to coordinate defect prevention activities exists
	A team to coordinate defect prevention activities for the software project exists
	Adequate resources and funding are provided for defect prevention activities at the project and organization levels
	Members of the software engineering group and other software-related groups receive required training to perform their defect prevention activities
Activities Performed	The software project develops and maintains a plan for its defect prevention activities
	At the beginning of a software task, the members of the team performing the task meet to prepare for the activities of that task and the related defect prevention activities
	Causal analysis meetings are conducted according to a documented procedure
	Each of the teams assigned to coordinate defect prevention activities meets on a periodic basis to review and coordinate implementation of action proposals from the causal analysis meetings
	Defect prevention data are documented and tracked across the teams coordinating defect prevention activities
	Revisions to the organization’s standard software process resulting from defect prevention actions are incorporated according to a documented procedure
	Revisions to the project’s defined software process resulting from defect prevention actions are incorporated according to a documented procedure
	Members of the software engineering group and software-related groups receive feedback on the status and results of the organization’s and project’s defect prevention activities on a periodic basis

Table 18. Defect Prevention

Measurement and Analysis	Measurements are made and used to determine the status of the defect prevention activities
Verifying Implementation	The organization’s activities for defect prevention are reviewed with senior management on a periodic basis
	The software project’s activities for defect prevention are reviewed with the project manager on both a periodic and event-driven basis
	The software quality assurance group reviews and/or audits the activities and work products for defect prevention and reports the results

Table 19. Technology Change Management

Goals	Incorporation of technology changes are planned
	New technologies are evaluated to determine their effect on quality and productivity
	Appropriate new technologies are transferred into normal practice across the organization
Commitment to Perform	The organization follows a written policy for improving its technology capability
	Senior management sponsors the organization’s activities for technology change management
	Senior management oversees the organization’s technology change management activities
Ability to Perform	A group responsible for the organization’s technology change management activities exists
	Adequate resources and funding are provided to establish and staff a group responsible for the organization’s technology change management activities
	Support exits for collecting and analyzing data needed to evaluate technology changes
	Appropriate data on the software processes and software work products are available to support analyses performed to evaluate and select technology changes
	Members of the group responsible for the organization’s technology change management activities receive required training to perform these activities
Activities Performed	The organization develops and maintains a plan for technology change management
	The group responsible for the organization’s technology change management activities works with the software projects in identifying areas of technology change
	Software managers and technical staff are kept informed of new technologies
	The group responsible for the organization’s technology change management systematically analyzes the organization’s standard software process to identify areas that need or could benefit from new technology

Table 19.   Technology Change Management

	Technologies are selected and acquired for the organization and software projects according to a documented procedure
	Pilot efforts for improving technology are conducted, where appropriate, before a new technology is introduced into normal practice
	Appropriate new technologies are incorporated into the organization’s standard software process according to a documented procedure
	Appropriate new technologies are incorporated into the projects’ defined software processes according to a documented procedure
Measurement and Analysis	Measurements are made and used to determine the status of the organization’s activities for technology change management
Verifying Implementation	The organization’s activities for technology change management are reviewed with senior management on a periodic basis
	The software quality assurance group reviews and/or audits the activities and work products for technology change management and reports the results

Table 20. Process Change Management

Goals	Continuous process improvement is planned
	Participation in the organization’s software process improvement activities is organization wide
	The organization’s standard software process and the projects’ defined software processes are improved continuously
Commitment to Perform	The organization follows a written policy for implementing software process improvements
	Senior management sponsors the organization’s activities for software process improvement
Ability to Perform	Adequate resources and funding are provided for software process improvement activities
	Software managers receive required training in software process improvement
	The managers and technical staff of the software engineering group and other software-related groups receive required training in software process improvement
	Senior management receives required training in software process improvement
Activities Performed	A software process improvement program is established which empowers the members of the organization to improve the processes of the organization
	The group responsible for the organization’s process activities (e.g., software engineering process group) coordinates the software process improvement activities
	The organization develops and maintains a plan for software process improvement according to a documented procedure
	The software process improvement activities are performed in accordance with the software process improvement plan
	Software process improvement proposals are handled according to a documented procedure
	Members of the organization actively participate in teams to develop software process improvements for assigned process areas
	Where appropriate, the software process improvements are installed on a pilot basis to determine their benefits and effectiveness before they are introduced into normal practice

Table 20. Process Change Management

	When the decision is made to transfer a software process improvement into normal practice, the improvement is implemented according to a documented procedure
	Records of software process improvement activities are maintained
	Software managers and technical staff receive feedback on the status and results of the software process improvement activities on an event-driven basis
Measurement and Analysis	Measurements are made and used to determine the status of the software process improvement activities
Verifying Implementation	The activities for software process improvement are reviewed with senior management on a periodic basis
	The software quality assurance group reviews and/or audits the activities and work products for software process improvement and reports the results